An Approximate Euclidean Distance Calculation for Fast VQ Encoding*

Seong-Joon Baek** · Jinyoung Kim** · Sangki Kang***

ABSTRACT

In this paper, we present a fast encoding algorithm for vector quantization with an approximate Euclidean distance calculation. An approximation is performed by converting floating point to the near integer. An inequality between the approximate Euclidean distance and the nearest distance is developed to avoid unnecessary distance calculations. Since the proposed algorithm rejects those codewords that are impossible to be the nearest codeword, it produces the same output as conventional full search algorithm.

Keywords: Approximate Euclidean Distance, Fast Encoding Algorithm, Vector Quantization

I. Introduction

Vector quantization(VQ) is a very efficient approach to low-bit-rate image compression[1]. VQ has the potential to achieve coding performance close to the rate-distortion limit with increasing vector dimension. However, the utilization of VQ is severely limited by its encoding complexity which increases exponentially with dimension. Thus, it is important to reduce the computational complexity of VQ for realizing the full potential of VQ. onsequently, many techniques have been developed for speeding the full(minimum-distortion) search of an arbitrary codebook[2].

Here we present a novel fast encoding algorithm which uses an Euclidean distance approximation via converting floating point to an integer. A new inequality between the approximate Euclidean distance and the nearest distance is derived for the algorithm. Since the vector searching area is reduced with the inequality, the encoding time of the proposed algorithm is reatly reduced.

II. The Proposed Algorithms

Usually, the codewords and the input vectors are expressed as floating point. If we could

---

* This work was supported in part by the Chonnam National University.
** Dept. of Electronics, Chonnam National University.
*** Samsung Electromics, co. ltd.

use an integer instead of floating point, overall encoding process would be accelerated since the integer operation is much faster than the floating point operation.

Here we use a truncation method for converting floating point to an integer, which is common for most of recent microprocessors. For example, a floating point number x_i is expressed as $n_{x_i} + \Delta_{x_i}$, where n_{x_i} is the nearest integer not greater than x_i and Δ_{x_i} is a residual term which is between 0 and 1. Now the distance $\sum_{i=1}^{k}(x_i - y_i)^2$ can be approximated as $\sum_{i=1}^{k}(n_{x_i} - n_{y_i})^2$. This approximation greatly accelerate the distance calculation. However it can not replace the original distance calculation because it inevitably introduces approximation error. Considering them, we will give a following Lemma which plays a key role in the proposed algorithm.

Lemma : Let $x = (x_1, x_2, \ldots, x_k)$ be an k dimensional test vector and $y = (y_1, y_2, \ldots, y_k)$ be a codeword.

$$\sum_{i=1}^{k}(x_i - y_i)^2 \geq \sum_{i=1}^{k} m_i^2,$$

where

$$m_i = \begin{cases} 0 & , n_{x_i} = n_{y_i} \\ n_{x_i} - n_{y_i} - 1 & , n_{x_i} > n_{y_i} \\ n_{y_i} - n_{x_i} - 1 & , n_{x_i} < n_{y_i} \end{cases}$$

Proof : The proof is very simple. Assuming that $n_{x_i} > n_{y_i}$,

$$\begin{aligned}(x_i - y_i)^2 &= (n_{x_i} + \Delta_{x_i} - n_{y_i} - \Delta_{y_i})^2 \\ &= (n_{x_i} - n_{y_i} + \Delta_{x_i} - \Delta_{y_i})^2,\end{aligned}$$

Since $-1 < \Delta_{x_i} - \Delta_{y_i} < 1$, the following inequality holds true.

$$(x_i - y_i)^2 > (n_{x_i} - n_{y_i} - 1)^2.$$

Since the proof of the other cases is also simple, we left them to the reader. The equality holds true only when every x_i is the same as y_i.

With the above Lemma at hand, now turn to describe the proposed algorithm. The algorithm consists of two steps. The first is a checking step whether distance calculation is needed or not. And the second step is an actual distance calculation step.

At the first step, for a given vector x and the codeword y, the algorithm calculates m_i^2 according to the Lemma and checks if

$$\sum_{i=1}^{k} m_i^2 \geq d^2_{\min},$$

where $d_{\min}$ is the minimum distance so far. In this step, it should be noted that m_i^2 can be precomputed and retrieved by table lookup. This means that no multiplications are needed in relation to the test. If the answer to the test is yes, the codeword is rejected without calculating the Euclidean distance. Otherwise, the distance is calculated.

Now inspect the number of operations required for the proposed algorithm in detail. Assume that the size of codebook is N and the number of needed actually distance calculations is M. Full search algorithm requires NK floating point multiplications and $N(2K-1)$ floating point additions(including subtractions), whereas the proposed algorithm requires MK floating point multiplications and $M(2K-1)$ floating point additions and $N(K-1)$ integer additions. Comparing them, the proposed algorithm saves $(N-M)K$ floating multiplications and $(N-M)(2K-1)$ floating point additions and loses $N(K-1)$ integer additions.

Consequently, the proposed algorithm reduces the number of multiplications and additions inversely proportional to the number of the required distance calculations. Approximation error can be expressed by the following way.

$$\begin{aligned}(x_i-y_i)^2-(n_{x_i}-n_{y_i})^2 &= 2(\Delta_{x_i}-\Delta_{y_i})(n_{x_i}-n_{y_i})+(\Delta_{x_i}-\Delta_{y_i})^2 \\ &\cong 2(\Delta_{x_i}-\Delta_{y_i})(n_{x_i}-n_{y_i}).\end{aligned}$$

According to the above results, approximation error is proportional to $n_{x_i}-n_{y_i}$. Usually the existing fast algorithm calculates the distance of the codeword which is in the vicinity of the test vector and discards the other codewords to reduce the required number of distance calculations. Considering them, we can say that the proposed algorithm, if combined with the fast algorithm, could be benefited from two facts. The first is the reduced number of distance calculations, which reduces the effective N. And the second is small approximation error, which reduces the effective M. In the next section, we will show the simulation results to confirm the effectiveness of the proposed algorithm.

III. Experimental results

Experiments have been performed using four USC images(Lena, Boat, Man, Baboon) to evaluate the efficiency of the proposed algorithm. The images are 512×512 monochrome with 256. 'Lena' image was used as the training set to generate the codebook with dimension 16(4×4).

Table 1. The average Number of Distance Calculations

Image	Encoding Method	The number of distance calculations		
Lena	Full	128	256	512
	Proposed	37.07	58.85	97.49
	ENNS	10.20	17.64	32.81
	Proposed+ENNS	2.36	2.85	3.39
Boat	Full	128	256	512
	Proposed	40.17	61.80	105.55
	ENNS	14.50	26.43	49.50
	Proposed+ENNS	2.51	2.95	3.62
Man	Full	128	256	512
	Proposed	32.15	49.34	80.73
	ENNS	13.72	25.10	46.55
	Proposed+ENNS	2.50	3.05	3.74
Baboon	Full	128	256	512
	Proposed	36.76	57.63	92.94
	ENNS	28.74	55.16	105.52
	Proposed+ENNS	3.08	3.75	4.52

In the Table 1, the average number of distance calculations is compared and shown for various codebook size. It is apparent that the average number of distance calculations of the proposed algorithm is only a small portion of full search algorithm.

Table 2. The Average Number of Distance Calculations For Speech Data

Data	Encoding Method	The number of distance calculations		
Speech 1	Full	128	256	512
	Proposed	36.61	55.25	110.21
	ENNS	22.66	41.57	73.89
	Proposed+ENNS	4.12	5.52	6.23
Speech 2	Full	128	256	512
	Proposed	32.14	45.37	98.45
	ENNS	21.48	40.25	72.15
	Proposed+ENNS	3.98	5.15	6.02

Speech data were also experimented with 8 bit PCM waveform and 8 dimension vector. Table 2 shows the results. The tables also show the results when the proposed algorithm is combined with the fast algorithm. ENNS algorithm[3, 4, 5] is chosen as a fast algorithm because it is rather simple to implement and is fast enough for general purpose. Apperently, the proposed algorithm combined with ENNS is shown to be the fastest one.

However the encoding time is not straightly dependent on the above results. The fast algorithm such as ENNS needs some calculation overhead to discard unnecessary codewords. This overhead should be taken into consideration when the proposed algorithm is combined with ENNS algorithm.

When the codebook size is 256 and Lena is an input image, average number of distance calculations is 2.85. When compared with the full search algorithm, the proposed algorithm saves 4050.4((256-2.85)×16) floating point multiplications and 7847.65((256-2.85)×(2×16-1)) floating point additions while it needs 264.5(17.64×(16-1)) integer additions, where 17.64 is the effective number of codewords. Additionally, some small computation overhead related to ENNS algorithm should be added. Since the execution time of floating point and integer multiplications and additions highly depends on the type of the processor, the exact savings of encoding time would be relative to the processor actually used.

IV. Conclusion

In this paper, we present a fast encoding algorithm for vector quantization with an approximate Euclidean distance calculation. An approximation is performed by converting floating point to an integer. An inequality between the approximate Euclidean distance and the nearest distance is developed to avoid unnecessary distance calculations. Experimental results confirm the effectiveness of the proposed algorithm. However, it should be noted that the proposed algorithm requires some additional storage.

References

[1] Gray, R. M. 1984. "Vector Quantization." *IEEE ASSP Magazine, vol. 1,* pp. 4-29, Apr. 1984.
[2] Gray, R. M. & Neuhoff, D. L. 1998. "Quantization." *IEEE Trans. Information Theory, vol 44,* no. 5, pp. 2325-2383, Oct.
[3] Ra, S. W. & Kim, J. K. 1993. "A fast mean-distance-ordered partial codebook search algorithm for image vector quantization." *IEEE Trans. Circuits Syst. II,* vol 40, no. 9, pp. 576-579, Sept.
[4] Baek, S. J., Jeon, B. K. & Sung, K.-M. 1997. "A Fast Encoding Algorithm for Vector

Quantization." *IEEE Signal Processing Letters, vol. 4,* no. 12, pp. 2-4, Dec.
[5] Baek, S. J., Bae, M. J., Sung, K.-M. 1999. "A Fast Vector Quantization Encoding Algorithm Using Multiple Projection Axes." *Signal Processing, vol. 75,* pp. 89-92.

접수일자: 2004. 4. 30
게재결정: 2004. 6. 15

▲ 백성준(Seong-Joon Baek)
광주광역시 북구 용봉동 300번지 (우: 500-757)
전남대학교 전자컴퓨터정보통신공학부
Tel: +82-62-530-1795 Fax: +82-62-530-1759
E-mail: tozero@chonnam.ac.kr

▲ 김진영(Jinyoung Kim)
광주광역시 북구 용봉동 300번지 (우: 500-757)
전남대학교 전자컴퓨터정보통신공학부
Tel: +82-62-530-1757 Fax: +82-62-530-1759
E-mail: beyondi@chonnam.ac.kr

▲ 강상기(Sangki Kang)
경기도 수원시 팔달구 매탄 3동 416 사서함 105
삼성전자 정보통신 총괄 통신연구소
Tel: +82-31-279-5486 Fax: +82-31-9530-4606
E-mail: sangki.kang@samsung.com